EECS 349 (Machine Learning) Homework 5

WHAT TO HAND IN

You are to submit the following things for this homework:

1. A **PDF** document containing answers to the homework questions.
2. The **WELL COMMENTED** source code for all software you write.

   NOTE: If your code is not well commented do not expect full points. Every function should have comments associated with it that specify what the input data format is, what the output data format is and what the function does. Any tricky bits in the code should have a comment explaining them

HOW TO HAND IT IN

To submit your assignment:

1. Compress all of the files specified into a .zip file.
2. Name the file in the following manner, firstname_lastname_hw5.zip. For example, Bryan_Pardo_hw5.zip.
3. Submit this .zip file via Canvas.

DUE DATE: the start of class as specified in the course calendar.

1) Boosting (3 points)

The AdaBoost algorithm is described in the paper "A Brief Introduction to Boosting" by Robert Schapire. You have been provided this paper on the course website. This algorithm learns from a training set of input/output instances $\{(x_1, y_1),...,(x_m, y_m)\}$ where X is an arbitrary set of inputs, $x_i \in X$ and $y_i \in \{-1,+1\}$. AdaBoost assumes the training set is fixed over all the rounds. Assume now that after each round, each pair in the training set has a probability q of reversing the sign of its label (the y term in the pair is the label).

A. (1 point) How would the performance of AdaBoost be affected by a dataset that changes from round to round in this manner?

B. (1 point) Assume that the number of rounds is fixed at some value T. Modify AdaBoost to improve its expected performance on the training set immediately after round T. Describe your modification in both text and mathematical notation.

C. (1 point) Give an informal argument for why your modification would do better than the unmodified AdaBoost. Illustrate with a simple example that you've worked through.

2) Active Learning (2 points)

Read the paper "Improving Generalization with Active Learning" (there is a link on the course calendar) and answer the following questions.

A.(1 point): Give a definition of active learning. Give a reason why someone would want to use active learning. Motivate this with an example problem that illustrates the issue.

B. (1/2 point) Explain how to use the set of hypotheses in a version space to select the next query for active learning. What is the name of this kind of query selection?

C. (1/2 pint) For an SG neural network on the 25-bit threshold problem, what function appears to govern the relationship between the number of samples and the error rate of the learner? How does this compare to using random sample selection?

3) Using a Hidden Markov Model (3 points)

Model 1

Starting Probability (π)

Down	Up	Same
0.3	0.3	0.4

Transition Probability (A)

	Down	Up	Same
Down	0.5	0.2	0.3
Up	0.5	0.2	0.3
Same	0.5	0.2	0.3

end state

Observation Probability (B)

	Coffee	OJ	Tea
Down	0.1	0.5	0.4
Up	0.5	0.3	0.2
Same	0.2	0.2	0.6

observation

Model 2

Starting Probability (π)

Down	Up	Same
0.5	0.3	0.2

Transition Probability (A)

	Down	Up	Same
Down	0.2	0.3	0.5
Up	0.6	0.2	0.2
Same	0.6	0.3	0.1

end state

Observation Probability (B)

	Coffee	OJ	Tea
Down	0.1	0.6	0.3
Up	0.6	0.3	0.1
Same	0.2	0.1	0.7

observation

The figure above shows two hidden Markov models for IBM's stock price. For both models, each hidden state indicates the difference between the IBM's starting value and ending value for a single day. Thus, "Down" means it was a day where the stock decreased. *NOTE* For the transition probability table, rows indicate starting states and columns indicate ending states.

Every morning, the president of IBM orders a drink on the way to work. Sometimes it is tea, sometimes coffee, sometimes orange juice (OJ). Over time both hidden Markov models were built by observing his morning drink purchase and then correlating them with the actual performance of IBM's stock.

Observation Sequence 1: {Coffee, Coffee, Coffee, Tea, Tea}

Observation Sequence 2: {OJ, Coffee, Tea, Coffee, Coffee}

Observation Sequence 3: {Tea, Coffee, Tea, Coffee, Tea, Coffee }

Assume you have witnessed Observation Sequence 1 and Observation Sequence 2. Determine whether Model 1 or Model 2 is more likely (assume the prior probability of each model is 0.5). Explain your reasoning and show your calculations. What is the name of the algorithm you used to calculate your results?

EECS 349 (Machine Learning) Homework 5

4) Reading up on Markov Models (2 points)

A. (1 point) Read the Rabiner paper "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" available on the course calendar. Give two approaches to modeling state duration for HMMs. Compare their strengths and weaknesses.

B. (1 point) Read the paper "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," available on the course website. Compare HMMs to Deep Neural Networks on the problem of speech recognition. What is the traditional approach that has been used for decades? What approach are the authors proposing to use? Is the new approach better? If so, in what way? Back up your assertion with experimental evidence.

5) Genetic Algorithm Design (5 points)

You must design (but don't have to implement) a genetic algorithm (GA) that evolves decision tree classifiers based on the Ivy League training data from the first homework assignment. What follows is an example 3-line input file for a target concept 'People accepted to an Ivy League School'

```
GoodGrades GoodLetters GoodSAT IsRich Scholarship ParentAlum SchoolActivities CLASS
true  true  true  true  true  true  true  true
true  true  true  false  true  true  false false
```

The first row gives the variable names. All variables are Boolean. Subsequent rows are individuals who have applied to Ivy League schools. The final column in each row is the true classification (i.e. "true" means "accepted to an Ivy.')

There are a number of important design questions that must be addressed in designing a GA to make decision trees. Describe your design choices.

A. (1/2 point) Define how a decision tree will be encoded (genotype). Traditionally GA's use a linear string of bits or linear vector of numbers for the genotype. However, if you wish to use a tree-based genotype structure – more akin to *genetic programming* than genetic algorithms -- that is acceptable, though you will need to specify appropriate genetic mutation and crossover operators (such as swapping subtrees).[1]

B. (1/2 point) Explain how your encoding maps into a decision tree. (How genotype maps to phenotype).

C. (1/2 point) Describe the hypothesis space enabled by this encoding. Are there hypotheses that cannot be represented using your encoding? Is it the full set of possible Boolean functions you could generate from this data?

D. (1/2 point) Compare the possible decision trees that can be represented by your encoding to decision trees created by the ID3 algorithm.

E. (1/2 point) Define a fitness function. Explain EXACTLY how you will calculate fitness for a member of the population. This will presumably involve the provided training data somehow.

---

[1] *Students interested in prior research on this topic, might like to peruse this conference paper (published in 2000), which used a tree-based genotype structure to evolve decision trees, and reported several favorable results compared against C4.5.*
http://www.gatree.com/data/GATreICTAI00.pdf

EECS 349 (Machine Learning) Homework 5

F. (1/2 point) Define how mutation works. Precisely define your probability of mutation formula. What kinds of mutation operators will you use? Bit flips? Swapping of positions? Incrementing/decrementing? Gaussian-based mutation? Something else?

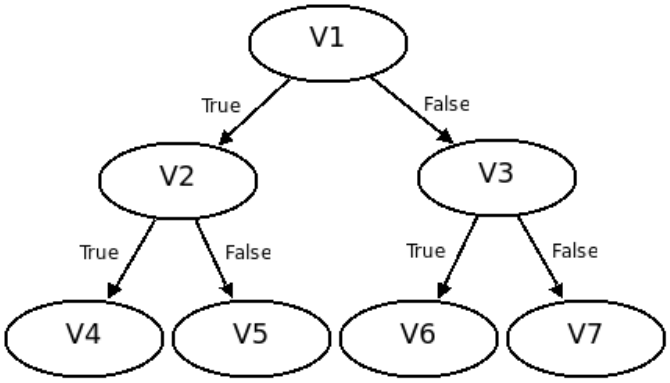
G. (1/2 point) Define how your crossover method works. One point? Multi-point? Uniform? Something else? Give an example.

H. (1/2 point) Define how selection works: How will you determine who reproduces? Roulette wheel/fitness proportional? Rank-based? Tournament-selection?

I. (1/2 point) Define the lifespan of population members. Do parents survive to the next generation to compete with children?

J. (1/2 point) Define a termination condition. When do you decide that the learner is finished? After a fixed number of generations? After a fixed percentage of the population reaches a certain fitness? When the diversity of the population decreases?

HINT



Because genotype-phenotype mapping can be a challenge, here is one possible encoding you may use. If you wish, you may also design your own.

A decision tree of maximum depth D can be encoded as a fixed-length vector ***V*** *(1-D array) of integers that is of length N, where* $N = (2^D - 1)$*, by using a bit of clever array indexing.*

$\boldsymbol{V} = V_1 \ldots V_N$*, where Vi has child nodes* V_{2i} *and* V_{2i+1}*, and it's parent node is* $V_{i/2}$ *(if you use integer division). This results in a tree that looks like the figure shown at right.*
Each node (V_i*) can be assigned an integer value of 0 ("return classification=false"), 1 ("return classification=true"), or one of the integers 2...K+1, where K is the number of features in your dataset. Each of 2...(K+1) represents a possible feature to split on. There is one subtle point about this representation: at the bottom layer of your tree further splitting must not be allowed, so you will want to convert any* V_i *value into either 0 or 1 (false or true classification), perhaps by value modulo 2.*